Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently amended) A method of authenticating a set of N information blocks, said method comprising:

obtaining an initial hash value for a set of N information blocks, wherein N is an integer;

altering one of said N information blocks from said set of N information blocks so as to form a revised set of N information blocks;

calculating a revised hash value for said revised set of N information blocks; while

calculating a check hash value for said N information blocks by performing a hash operation on said N information blocks; then

comparing said check hash value with said initial hash value; and

accepting said revised hash value for said revised set of N information blocks if said check hash value matches said initial hash value.

2. (original) The method as described in claim 1 wherein said calculating said revised hash value while calculating said check hash value comprises:

calculating said revised hash value in parallel with said check hash value.

3. (original) The method as described in claim 1 wherein said calculating said revised hash value while calculating said check hash value comprises:

hashing said altered block of data so as to obtain a first hashing result;

storing said first hashing result in a processor; and then

hashing the corresponding unaltered block of data so as to obtain a second hashing result.

4. (original) The method as described in claim 1 wherein said calculating said revised hash value while calculating said check hash value comprises:

concurrently hashing said altered block of data so as to obtain a first hashing result and hashing the corresponding unaltered block of data so as to obtain a second hashing result.

5. (original) The method as described in claim 1 wherein said calculating said revised hash value while calculating said check hash value comprises:

utilizing a single processor to calculate said revised hash value and to calculate said check hash value.

6. (original) The method as described in claim 1 and further comprising:

performing a linear hash of said set of data by hashing said N blocks of data in sequential order from block 1 to block N.

7. (original) The method as described in claim 1 wherein said obtaining said initial hash value for said set of N information blocks comprises:

hashing each of said N information blocks in said set of N information blocks.

8. (original) The method as described in claim 1 and further comprising:

storing said initial hash value in a processor.

9. (original) The method as described in claim 1 wherein said altering one of said N information blocks comprises:

storing a new value for at least part of one of said N information groups.

10. (original) The method as described in claim 1 wherein said comparing said check hash value with said initial hash value comprises:

determining whether said check hash value and said initial hash value are exactly the same.

11. (original) The method as described in claim 1 wherein said accepting said revised hash value comprises:

replacing said initial hash value with said revised hash value.

12. (original) The method as described in claim 1 and further comprising:
storing the new revised hash value in the memory area previously occupied by the initial hash value.

13. (original) The method as described in claim 1 and further comprising:
not accepting said revised hash value as a replacement for said initial hash value if said check hash value does not match said initial hash value.

14. (original) The method as described in claim 13 and further comprising:
indicating a failure to authenticate.

15. (original) The method as described in claim 1 and further comprising:
utilizing said set of data for digital rights management.

16. (original) The method as described in claim 1 and further comprising:
replacing said initial hash value with said revised hash value.

17. (original) The method as described in claim 1 and further comprising:
receiving as part of an initialization routine a length of a data set to be hashed, wherein said data set is comprised of said N information groups.

18. (original) The method as described in claim 17 and further comprising:
padding at least one of said N information groups so that each of said N information groups is of equal length.

19. (original) The method as described in claim 1 and further comprising:
initializing a processor so as to perform a hashing routine.

20. (original) The method as described in claim 1 and further comprising:
initializing a hashing routine by entering the length of said set of data.

21. (original) The method as described in claim 1 and further comprising:
dividing the set of data into a plurality of blocks.

22. (original) The method as described in claim 1 and further comprising:
dividing the set of data into a plurality of blocks of data;
padding the last block of data so that each of said blocks of data is of equal length.

23. (original) A method of authenticating a set of N information blocks, said method comprising:

obtaining an initial root key for a set of data comprised of a plurality of blocks of data, said root key operable for authenticating said set of data;

calculating hash keys for said plurality of blocks of data so that each of said hash keys corresponds to only one of said blocks of data and so that each of said blocks of data corresponds to only one of said hash keys;

storing said hash keys for said plurality of blocks of data;

altering one of said blocks of data so as to form a revised block of data;

calculating a second hash key for said revised block of data, wherein said revised block of data immediately prior to being revised corresponds to a first hash key and wherein said first hash key is one of said hash keys for said plurality of blocks of data;

utilizing said stored hash keys, including said first hash key, to calculate a check root key while utilizing said stored hash keys and said second hash key substituted in place of said first hash key to calculate a new root key;

comparing said check root key with said initial root key;

accepting said new root key if said check root key matches said initial root key.

24. (original) The method as described in claim 23 wherein said utilizing said stored hash keys, including said first hash key, to calculate said check root key is done in parallel with said utilizing said stored hash keys and said second hash key substituted in place of said first hash key to calculate said new root key.

25. (original) The method as described in claim 24 and further comprising:

computing a branch key;
hashing said branch key and said first hash key; and
hashing said branch key and said second hash key.

26. (original) The method as described in claim 24 and further comprising:
computing a branch key;
hashing said branch key and said first hash key; while
hashing said branch key and said second hash key.

27. (original) The method as described in claim 24 and further comprising:
computing a branch key; and concurrently
computing a result from said branch key and said first hash key; while
computing a result from said branch key and said second hash key.

28. (original) The method as described in claim 24 and further comprising:
utilizing a single processor to calculate said check root key and said new root key.

29. (original) The method as described in claim 23 and further comprising:
dividing an initial set of data into X blocks, where X is equal to 2 raised to the Y power and where Y is an integer.

30. (original) The method as described in claim 23 and further comprising:
calculating intermediate branch keys by hashing previously determined branch keys; and then
utilizing said intermediate branch keys to calculate said new root key.

31. (original) The method as described in claim 23 and further comprising:
encrypting said hash keys for said plurality of blocks; and
storing said encrypted hash keys in memory outside of a processor.

32. (original) The method as described in claim 23 and further comprising:

storing said hash keys for said plurality of blocks in a processor.

33. (original) The method as described in claim 23 and further comprising:
storing said root key inside a processor.

34. (previously presented) The method as described in claim 23 wherein said altering one of said blocks of data comprises:
storing a new value for at least part of one of said information groups.

35. (original) The method as described in claim 23 wherein said comparing said check root key with said initial root key comprises:
determining whether said check root key and said initial root key are exactly the same.

36. (original) The method as described in claim 23 wherein said accepting said new root key comprises replacing said initial root key with said new root key.

37. (original) The method as described in claim 36 and further comprising:
storing said new root key in a processor in a memory area previously occupied by said initial root key.

38. (original) The method as described in claim 23 wherein said set of N information blocks is at least partially utilized for managing digital rights.

39. (original) The method as described in claim 23 wherein said set of N information blocks is at least partially utilized as an entitlement control message for receiving a program.

40. (original) The method as described in claim 23 and further comprising:
initializing a hashing function by receiving the length of said N information blocks.

41. (original) The method as described in claim 40 and further comprising:
padding the final block of the N information blocks prior to hashing the Nth block.

42. (original) The method as described in claim 23 and further comprising:
initializing a hashing function.

43. (original) The method as described in claim 23 and further comprising:
obtaining a set of data; and
dividing said set of data into a plurality of blocks.